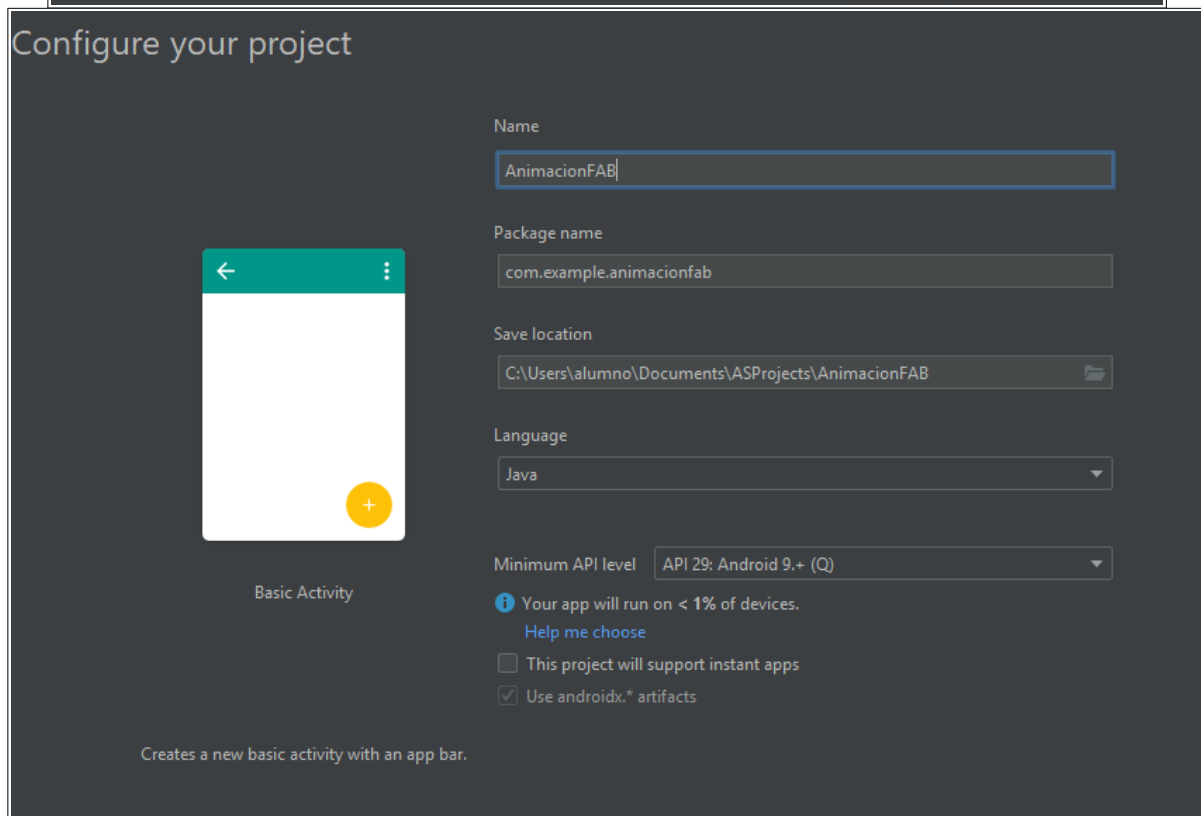
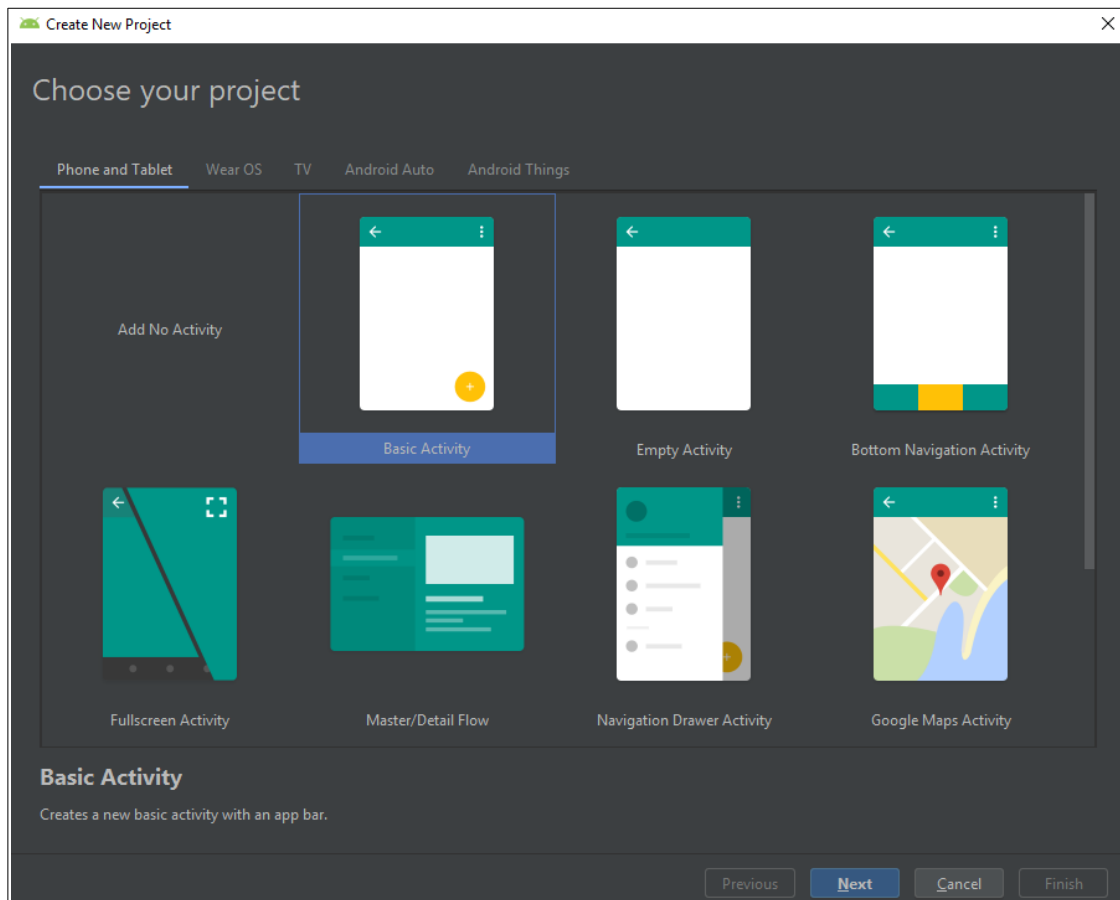
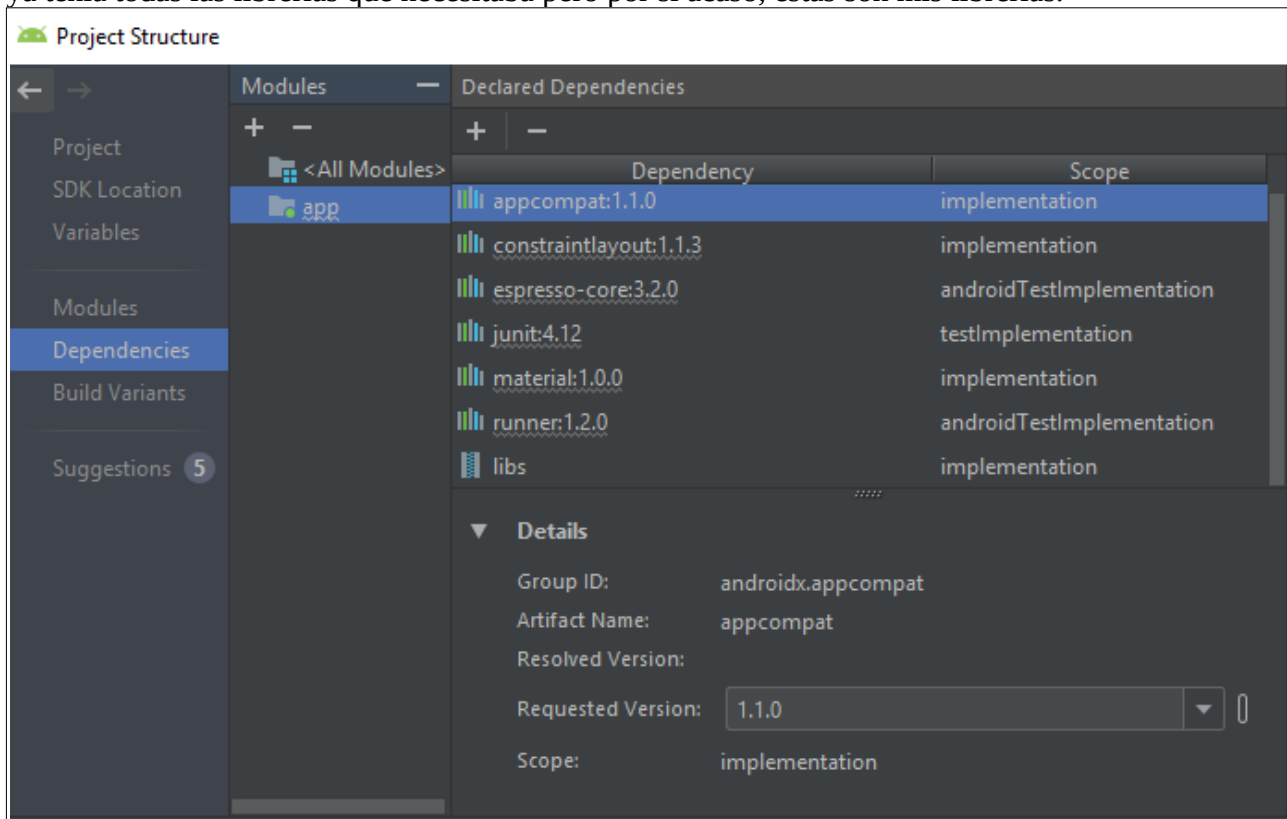


Lo primero, para implementar el código, es necesario crear un nuevo proyecto en Android Studio, con una “Basic Activity” ya que trae un Floating Action Button implementado de serie.

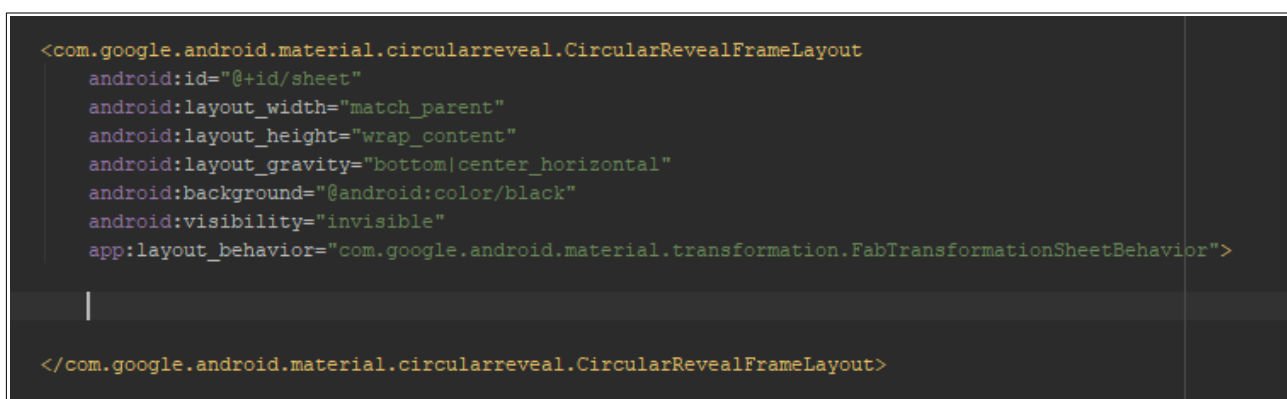


Lo siguiente que vamos a hacer es revisar las librerías que tenemos en el proyecto, yo por defecto ya tenía todas las librerías que necesitaba pero por si acaso, estás son mis librerías.



Una vez revisado lo anterior, vamos a revisar el código XML del “activity_main.xml”. Para poder implementar la animación del FAB, es necesario un layout especial:

`com.google.android.material.circularreveal.CircularRevealFrameLayout`



Dentro de este layout, podemos implementar un layout que contendrá todos los elementos de nuestro objeto que aparece y desaparece, en este caso es un Toolbar con 4 opciones que consisten en 4 ImageViews.

```
<LinearLayout
    android:id="@+id/barra"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:layout_alignParentBottom="true"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/uno"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:scaleType="centerInside"
        android:src="@drawable/abc_ic_menu_copy_mtrl_am_alpha" />

    <ImageView
        android:id="@+id/dos"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:scaleType="centerInside"
        android:src="@drawable/abc_ic_menu_paste_mtrl_am_alpha" />
```

```
    <ImageView
        android:id="@+id/tres"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:scaleType="centerInside"
        android:src="@drawable/abc_ic_menu_selectall_mtrl_alpha" />

    <ImageView
        android:id="@+id/cuatro"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:scaleType="centerInside"
        android:src="@drawable/abc_ic_menu_cut_mtrl_alpha" />

</LinearLayout>
```

Después de colocar nuestro layout animado, tenemos que implementar el FAB, que por ser una Basic Activity ya estaba implementado en el proyecto, por si acaso este es el código XML del FAB:

```
<!-- FAB -->

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:layout_gravity="end|bottom"
    android:src="@android:drawable/ic_input_add" />
```

Una vez realizado lo anterior, ya tendríamos nuestro XML completo.

Ahora pasamos a ver el código java. Para poder acceder al FAB desde cualquier lugar, lo hacemos un atributo de clase que inicializamos en el método principal “onCreate”.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    private FloatingActionButton fab;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        fab = findViewById(R.id.fab);
    }
}
```

En el código java lo único que tenemos que hacer para implementar la animación es implementar un OnClickListener en el FAB y en los elementos que aparecen en la Toolbar; para los elementos de la barra utilizaremos la herencia de la interfaz y para el FAB utilizaremos un listener anónimo.

```
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        fab.setExpanded(true);
    }
});
```

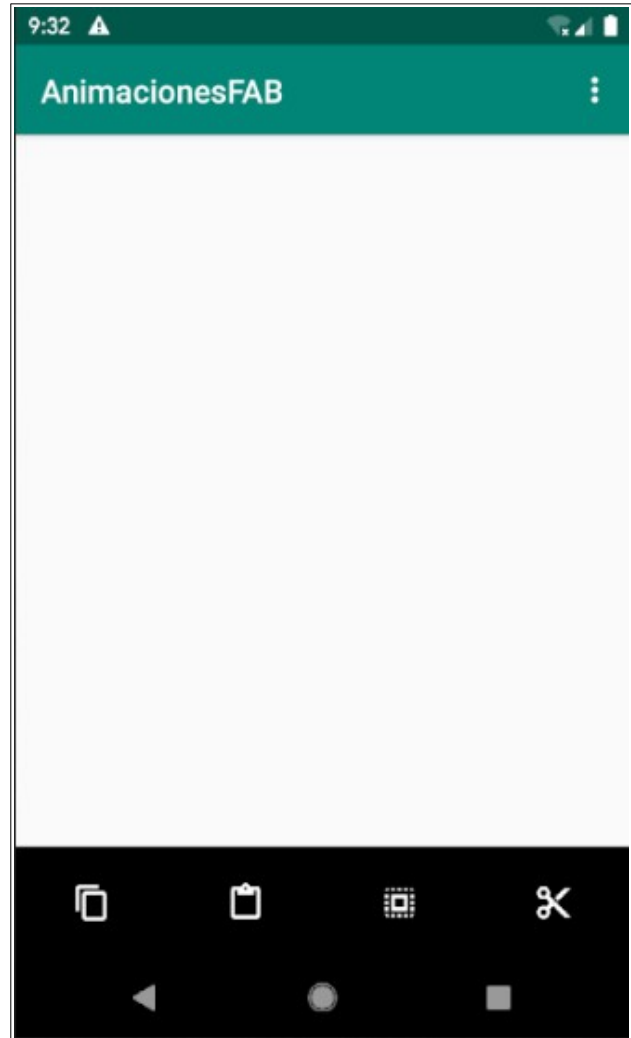
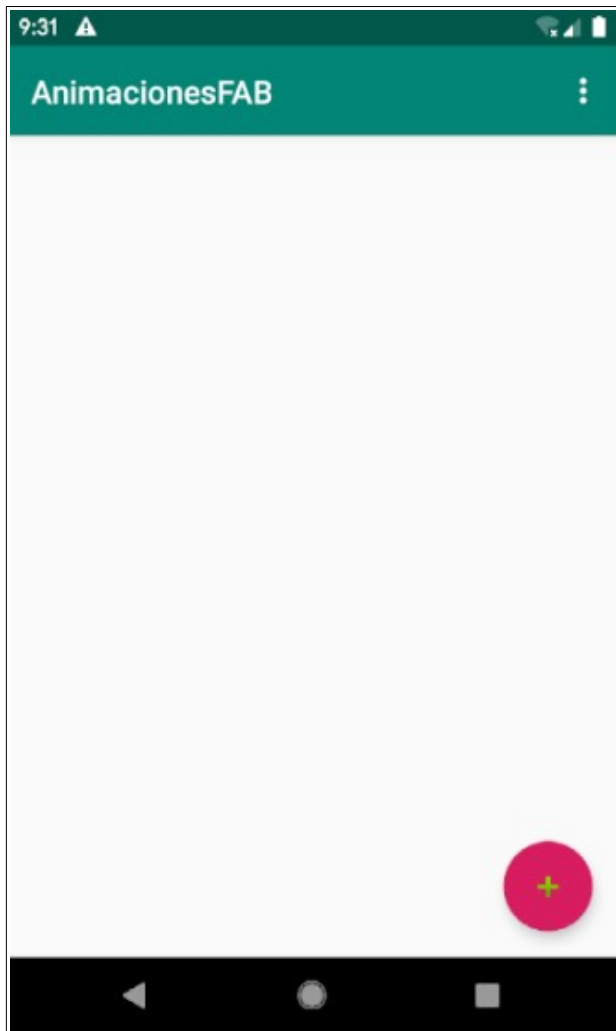
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    fab = findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            fab.setExpanded(true);
        }
    });

    View uno = findViewById(R.id.uno);
    View dos = findViewById(R.id.dos);
    View tres = findViewById(R.id.tres);
    View cuatro = findViewById(R.id.cuatro);

    uno.setOnClickListener(this);
    dos.setOnClickListener(this);
    tres.setOnClickListener(this);
    cuatro.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    fab.setExpanded(false);
    Toast.makeText(context, this, "Has pulsado la barra :", Toast.LENGTH_SHORT).show();
}
```

Cada vez que pulso en el FAB, pongo la propiedad “Expanded” con el método “setExpanded” a “false”, a su vez cada vez que pulso en alguna de las cuatro opciones de la Toolbar, pongo la propiedad antes mencionada a “true”, además de mostrar un pequeño mensaje opcional.



Esto provoca que aparezca o desaparezca la Toolbar.